

**PATENT**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re the Application of:	)	Examiner: Nirav B. PATEL
	)	
Jean-Marie GATTO	)	Art Unit: 2135
Thierry BRUNET DE COURSSOU	)	
	)	Confirmation No.: 9438
Serial No.: 10/789,975	)	
	)	Customer No.: 86915
Filed: February 27, 2004	)	
	)	
For: <b>DYNAMIC CONFIGURATION</b>	)	
<b>OF A GAMING SYSTEM</b>	)	
	)	<b><u>APPEAL BRIEF</u></b>
	)	

Atty. Docket No.: CYBS5858

---

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P. O. Box 1450  
Alexandria, VA 22313-1450

Sirs:

The present Appeal Brief appeals the final Office Action mailed November 24, 2009, and is submitted herewith pursuant to MPEP §1205.02. The Table of Contents may be found beginning on the following page.

The Director is hereby authorized to charge any fees due herewith under 37 C.F.R. §1.136(a) as well as the fees due under §41.20(b)(2), Fee Code 1402, in the amount of \$540.00, to Deposit Account No. 50-3159, referencing Docket No. CYBS5858.

## TABLE OF CONTENTS

<u>NO.</u>	<u>DESCRIPTION</u>	<u>PAGE</u>
I.	REAL PARTY IN INTEREST .....	3
II.	RELATED APPEALS AND INTERFERENCES .....	4
III.	STATUS OF CLAIMS .....	5
IV.	STATUS OF AMENDMENTS .....	6
V.	SUMMARY OF CLAIMED SUBJECT MATTER.....	7
	Independent Claim 17 .....	11
	Dependent Claims 18, 19 .....	13
	Independent Claim 20 .....	13
	Dependent Claim 21 .....	15
	Independent Claims 22 .....	15
	Dependent Claims 23 .....	17
	Independent Claims 24 .....	17
	Independent Claims 25 .....	19
VI.	GROUND FOR REJECTION TO BE REVIEWED ON APPEAL .....	21
VII.	ARGUMENTS.....	22
	Independent Claim 25 .....	22
	Independent Claim 17 .....	23
	Independent Claim 20 .....	29
	Independent claim 24.....	36
	Independent Claim 22 .....	43
III.	CLAIMS APPENDIX.....	48
IX.	EVIDENCE APPENDIX.....	59
X.	RELATED PROCEEDINGS.....	60

**I. REAL PARTY IN INTEREST**

The real party in interest is Mudalla Technology, Inc., a NY corporation having a principal place of business at c/o Thoits, Love, Hershberger & McLean, P.C., 245 Lytton Avenue, Suite 300 Palo Alto, CA 94301-1426. Mudalla Technology, Inc. is the new name of Cyberview Technology Inc., which change of name was recorded with the Delaware Secretary of State on July 10, 2008 and recorded at the USPTO on Aug. 14, 2010 at Reel/Frame 024837/0422. In turn, Cyberview Technology was the new name of Cyberscan Technology, Inc., which change of name was recorded at the USPTO on Feb. 20, 2007 and recorded at the USPTO at Reel/Frame 018907/0367. Cyberscan Technology, Inc. obtained the entire right, title and interest in and to the present application by virtue of an Assignment from the inventors to Cyberscan Technology, Inc. executed on June 9, 2004 and recorded at the USPTO at Reel/Frame 014713/0877.

II. RELATED APPEALS AND INTERFERENCES

None.

### **III. STATUS OF CLAIMS**

Claims 1-25, 82 and 84-90 are pending in the application. Of these, claims 1-16, 82 and 84-90 are withdrawn from consideration. The remaining claims 17-25 stand rejected. The rejection of claims 17-25 is appealed herewith.

#### **IV. STATUS OF AMENDMENTS**

No amendments were entered after the final Office Action mailed November 24, 2010.

A Request for Reconsideration was filed on March 22, 2010. An Advisory Action was mailed on April 6, 2010. A Notice of Appeal and a Pre-Appeal Brief Request for Reconsideration were filed on April 24, 2010. The Notice of Panel Decision from Pre-Appeal Brief Review notifying the applicant that the rejection of claims 17-25 would be maintained and that appeal would proceed to the Board of Patent Appeals and Interferences was mailed on June 1, 2010.

## **V. SUMMARY OF CLAIMED SUBJECT MATTER**

Many executable software components of regulated gaming machines are subject to receiving certification from a certification laboratory. According to the claimed embodiments of the present inventions, a unique PKI certificate is generated for each of the executable game components (subject to receiving certification) of each gaming machine and each executable game component that is subject to receiving certification is code signed with the generated PKI certificate. Each of the executable software components subject to receiving certification includes a unique identifier. Intrinsic information that uniquely identifies each executable software component may be obtained, for example, from various combinations of assembly directives and file property fields.

Fig. 3 illustrates the Certificate “Issued to” Field and shows the information that uniquely identifies each executable software component and that may be used to generate the “Subject Name” (or “Issued to” field) of the unique PKI certificate associated with each executable software component. The certificate authority responsible for generating the PKI certificate is shown in the “Issued by” field. Intrinsic information that uniquely identifies each executable software component may be entered in the extended attributes of a PKI certificate as an alternative to entering the information in the certificate Subject Name. In the same manner, additional identification information to those entered in the Subject Name may be entered in the extended attributes.

It may be determined whether the game software of gaming machines may be trusted by using, e.g., a Trusted Inventory tool that determines whether each executable software component subject to receiving certification is signed by a valid PKI certificate and its executable binary data is uncorrupted (its recalculated hash matches the code signature).

Not only must each executable game software component subject to receiving certification by a game certification lab be signed by a valid PKI certificate and its executable binary data uncorrupted, each must also be associated with a software restriction policy (SRP) certificate rule to allow execution of only those executable software components whose code signed PKI certificate is determined to be authorized. Setting the certificate subject name with a security level of "Unrestricted" ensures that only the executable software component identified in the certificate subject name is authorized to execute when the policy is enforced. SRP path rules must be configured such as to prevent non-authorized software from executing.

The signed executable software components may be packaged in code-signed MSI installation packages signed in a manner substantially identical to the executable software components, that is, with a unique PKI certificate whose subject name contains e.g., the part number, version and friendly name identifiers for the MSI package. Software Installation Policies (SIPs) may control the installation of each new game and are configured to automatically execute the MSI installation packages upon policy enforcement, provided the corresponding SRPs have been configured to authorize the execution of the MSI installation packages. If no SRP authorizes the execution of the MSI installation packages, the installation is ignored. When the MSI installation package is authorized to execute, the software components and other files contained in the package may be copied to the gaming terminals.

Game activation that authorizes execution of the game may be achieved by enforcing the associated SRPs. In the same manner, scheduled game activation and deactivation in order to offer selected authorized games to the players at predetermined authorized times may be achieved by simply enforcing the associated SRPs at a given time; this may be accomplished by having an operator manually enforce the SRP at a predetermined time via the group policy



management console, or having a process automatically enforce the SRP at a predetermined time via the API to the group policy management console. Alternatively, a selected executable software component may be prevented from executing by configuring its associated SRP security level to “disallowed”.

When a player selects a game from a gaming terminal from a game selection menu and requests execution thereof, the SRP of each executable game component subject to receiving certification is enforced. Indeed, the authenticode of the game's executable software components are verified by the associated enforced SRP before beginning execution. Should the authenticode verification fail, the gaming terminal may be locked pending servicing by an attendant. If the code is trusted, as verified by the associated enforced SRP, the game is allowed to execute.

A global verification process (Fig. 9) may be performed by a terminal to check that no unauthorized files are allowed to execute or affect the game outcome. The process may start with the gaming machine rebooting such that the operating system trusted base may be thoroughly verified before the game software components are verified. On completion of the operating system boot-up, the global verification process may be executed. The global verification process verifies all the executable files in given folder trees for trustworthiness. If any file is found to be untrusted, then the gaming machine may be frozen pending examination by security personnel. If the authenticode of all the files is trusted then the global verification process returns a trusted status. Consequently, all of the executable software components may be considered to be trusted.

However, it is to be noted that the fact that an executable software component is trusted does not imply that the software component is authorized to execute; it merely indicates that the software executable software component has a valid authorized authenticode certificate and that the software component binary data is not corrupted. Checking whether an executable software

component having a valid authorized authenticode certificate is authorized to execute requires that the applicable Software Restriction Policy be checked. This may be performed automatically when the software component is loaded by the operating system to start its execution, either when dynamically building the menu of authorized games, or each time upon starting execution of the game when the player has selected a game to play — or using an appropriate service that may be called by an application.

The three parties involved in a game cycle, according to embodiments of the present invention, are the game developer, the certification laboratory and the gaming operator. The game developer supplies the certification lab with the software components to be tested. The certification lab then certifies the supplied software components and provides the game developer with the certified software components for deployment across gaming machines. The authorized software components that were tested and certified by the certification lab may then be provided to the game operator (e.g., the casino) for installation and deployment on multiple gaming machines (such as GM001, GM002, GM2995 in Fig. 10) The certified authorized software components are code-signed using a certificate produced in accordance with an embodiment of the present invention, as described hereinabove. In this manner, since the same code-signed, authorized software components are provided to gaming operators for installation and deployment on multiple gaming machines, it follows that identical authorized software components in different ones of the gaming machines would be code signed with identical PKI certificates. This means that that non-identical authorized software components in different ones of the constituent computers would be code signed with separate and different PKI certificates, and means that no two non-identical authorized software components in different ones of the

constituent gaming machines are code signed with a same PKI certificate (since each executable game software component subject to certification is code signed with a unique PKI certificate).

Fig. 15 shows the enforcement of a Software Installation Policy (SIP). In Fig. 14, banks of gaming terminals are configured to execute authorized games using SIPs and SRPs policies. In order for the gaming terminals to be able to install a game, the associated SIP must be enforced. Fig. 16 illustrates the enforcement of a SRP. In Fig. 14, banks of gaming terminals are configured to execute authorized games using SIPs and SRPs. In order for the gaming terminals to be able to execute the games, the software restriction and installation policies must be enforced.

The embodiments described above, therefore, define methods that enable dynamic configuration of gaming machines by allocating an individual PKI certificate to each executable software component and each of its versions, binding the PKI certificate to the executable software, associating a distinctive policy for each certificate and then enforcing the software restriction policies.

#### **Independent Claim 17 and its dependent claims**

**Claim 17** defines a method for a network connected gaming system to prevent unauthorized software components of constituent computers of the gaming system from executing. The gaming system includes a plurality of gaming machines (see, e.g., GM001, GM002 ... GM295 of Fig. 10) each having a plurality of executable software components. The methods includes a step of producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, as disclosed in, for example, paragraphs [0045], [0047], [0118], [0120] and

**the originally-filed claims.** Each software component subject to receiving certification includes a unique identifier, as disclosed in paragraphs [041] – [043] and [047]. The method includes a step of code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate, as disclosed in paragraphs [045], [084], and **the originally filed claims.** Each of the respective PKI certificates, as claimed, is uniquely identified at least by a unique identifier that is uniquely associated with the executable software component. For example, as shown in Fig. 3, information (such as test certificate indicator 318, project/product code 320, type of executable code 322, part number 324, major/minor/build/version 326, certification lab identifier 328, friendly name 330) that uniquely identifies each executable software component, may be used to generate the “Subject Name” 316 (or “Issued to” field 306, 314, or also known as the “CommonName” field) of the individual PKI certificate 304 associated with each executable software component, thereby uniquely identifying respective PKI certificates with a unique identifier that is uniquely associated with a particular executable software component as disclosed, for example, in paragraph [045].

The certified software components, each code signed with a PKI certificate that is uniquely identified with a unique identifier associated with the executable software component, are distributed from the certification lab to the casino operators for deployment across estates of gaming machines, see paragraph [071]. This deployment means that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, and consequently means that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates. From

the foregoing, it also follows that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate, as claimed in claim 17. Lastly, claim 17 calls for configuring a software restriction policy (SRP) certificate rule for each of the plurality of executable software components and enforcing each of the software restriction policy certificate rules to allow execution of only those executable software components whose code signed PKI certificate is determined to be authorized, as disclosed in, for example, paragraphs [012], [015], [0115], [0118], [0119], **in the Summary section and in the originally-filed claims.**

**Claim 18** recites a further step of configuring software restriction policy rules to prevent execution of unauthorized software components, as disclosed, for example, in paragraphs [052], [054], [059], [064], [099], and throughout the specification.

**Claim 19** specifies that a step may be carried out of configuring software restriction policy rules to prevent execution of all not explicitly authorized software components. This subject matter is disclosed originally filed claim 17 and in the specification at paragraphs [0119], which states that SRPs may be relied upon to prevent non-authorized software components (i.e., “not explicitly authorized software components”), paragraph [121] and [0126].

**Independent claim 20 and its dependent claims**

Claim 20 defines a method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute. The method includes steps of code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized

software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate. The certified software components, each code signed with a PKI certificate that is uniquely identified with a unique identifier associated with the executable software component, are distributed from the certification lab to the casino operators for deployment across estates of gaming machines, see paragraph [071]. This deployment means that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, and consequently means that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates, which is really the same thing stated in three different ways.

Claim 20 then calls for the configuration of a separate software restriction policy for each authorized software component in each of the constituent computers of the gaming system, and associating the configured separate software restriction policy with the PKI certificate with which the authorized software component was code signed. These steps and functionality are disclosed in the specification at [012], [015], [0115] (“...Software Restriction Policy (SRP) configured with an individual PKI certificate associated to each authorized software component offer a “Policy/Enforce” model, or in other words a “Configure the Policy and then Enforce the Policy” model to enable network installation (or “game download”) and activation at predetermined times (or “game scheduling”) of selected authorized software components, in

order to control the software of the network connected gaming system and offer selected games to players.”), [0118], [0119], in the **Summary section and in the originally-filed claims.**

Lastly, Claim 20 calls for enforcing the associated software restriction policy for each code signed authorized software component such that each code signed authorized software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy, as disclosed in paragraphs [011], [015], [052], [054], [056], [057], [0101], Fig. 16, [0106], [0107], and **throughout the specification.**

**Claim 21** calls for the authorized software components to be mandated by a regulatory body, as disclosed in paragraph [011], [016] and throughout the specification wherein the **certification laboratories** are discussed relative to certifying software components for use in gaming machines.

**Independent claim 22 and its dependent claims**

**Claim 22** defines a method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute. The method includes steps of configuring a separate and unique certificate software restriction policy for each authorized executable software component of each of the constituent computers of the gaming system. These steps and functionality are disclosed in the specification at [012], [015], [0115]. The claim also requires that each authorized executable software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy, as disclosed in paragraphs [011], [015], [052], [054], [056], [057], [0101], Fig. 16, [0106], [0107], and **throughout the specification.**

Claim 22 then calls for a step of code signing each authorized software component with a PKI certificate; see the **Summary** section, **originally-filed claims** and paragraphs [010] (see discussion of code signing operation “signcode.exe” that bind the software component to the unique certificate), [045] and [084]. This code signing is carried out such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate. Indeed, the certified software components, each code signed with a PKI certificate that is uniquely identified with a unique identifier associated with the executable software component, are distributed from the certification lab to the casino operators for deployment across estates of gaming machines, see paragraph [071]. This deployment (the sending and installing of the same software in multiple gaming machines) means that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, and consequently means that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates.

Claim 22 then requires that a path software restriction policy be configured to prevent unauthorized and non-explicitly authorized software components from executing. This step and functionality is disclosed in the specification at **Fig. 7** and at paragraphs [0049], and in the **originally-filed claims**. The method then calls for enforcing the certificate software restriction



policy configured for each of the code signed authorized executable software components of each of the constituent computers of the gaming system, and enforcing the path software restriction policies, as disclosed in paragraphs [06], [09], [015] discussing the “policy/Enforce” model of the present inventions, Figs. 15-18, [049], [052], [053], [054], [045], [084], [057], [088], [101], [105] – [0110], and the **close-loop enforcement of policies**, disclosed beginning at paragraph [0116].

**Claim 23** calls for the authorized software components to be mandated by a regulatory body, as disclosed in paragraph [011], [016] and throughout the specification wherein the **certification laboratories** are discussed relative to certifying software components for use in gaming machines.

**Claim 24** defines a method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute. The gaming system is recited to include a plurality of gaming machines (see, e.g., GM001, GM002 ... GM295 of **Fig. 10**), each having a plurality of executable software components. The claimed method includes a step of producing a separate and unique PKI certificate for each of the plurality of executable software components within the gaming system subject to receive certification as disclosed in, for example, paragraphs [0045], [0047], [0118], [0120] and the **originally-filed claims**.

As with the above-discussed independent claims, each respective PKI certificate is then recited to be associated with a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical

executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate. As above, the certified software components, each code signed with a PKI certificate that is uniquely identified with a unique identifier associated with the executable software component, is distributed from the certification lab to the casino operators for deployment across estates of gaming machines, see paragraph [071]. This deployment (the sending and installing of the same software in multiple gaming machines) means that the same executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with the same identifiers and are code signed with the same PKI certificates, and consequently means that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates.

Each software component subject to receive certification is then code-signed with its respective separate and unique PKI certificate, as disclosed in the **Summary** section, **originally-filed claims** and paragraphs [010] (see discussion of code signing operation “signcode.exe” that bind the software component to the unique certificate), [045] and [084]. The claim’s steps then conclude with configuring (see [012], [015], [0115], [0118], [0119], in the **Summary** section and in the **originally-filed claims**) and enforcing a certificate software restriction policy for each of the respective separate and unique PKI certificates (See paragraphs [012], [015], [0115], [0118], [0119], in the **Summary** section and in the **originally-filed claims**).

Lastly, **claim 25** on appeal defines a method for downloading authorized executable software components and allowing execution of downloaded authorized executable software components of a plurality of gaming machines of a network connected gaming system (see, e.g., GM001, GM002 ... GM295 of **Fig. 10**). The recited method steps are carried out for each of the plurality of gaming machines of the network connected gaming system. These steps include a step of code signing each authorized executable software component with a separate PKI certificate that is unique to the authorized software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different gaming machines are code signed with a same PKI certificate. As above, the certified software components, each code signed with a PKI certificate that is uniquely identified with a unique identifier associated with the executable software component, is distributed from the certification lab to the casino operators for deployment across estates of gaming machines, see paragraph [071]. This deployment (the sending and installing of the same software in multiple gaming machines) means that the same executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with the same identifiers and are code signed with the same PKI certificates, and consequently means that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates.

The code signed authorized software components are then packaged into an installation package, as disclosed in the specification at [011], [012], [050], [052], [053], [065], [090], [099], [102] and [103], and corresponding **Figs.** The (software) install policies (or SIPs, in the specification) are then configured to install each code signed authorized executable software component contained in the installation package, as disclosed in the specification at paragraphs [052], [053], [088], [089], [095], [096], [099], [0101], [0115] and [0125], and corresponding **Figs.** Once the software components of the installation package are installed according to the install policies, certificate rule policies are configured and enforced to allow execution of the installed code signed authorized executable software component, as disclosed in paragraphs [012], [015], [0115], [0118], [0119], in the **Summary section and in the originally-filed claims.**

**VI. GROUND FOR REJECTION TO BE REVIEWED ON APPEAL**

- Whether claims 17-21, 24 and 25 are unpatentable under 35 USC §103(a) over Gunyakti et al. in view of Yip et al. in view of Fierres et al. in view of Lambert et al.

- Whether claims 22 and 23 are unpatentable under 35 USC §103(a) over Lambert et al. in view of Gunyakti et al. in view of Yip et al.

## VII. ARGUMENTS

### I. Independent claim 25 has yet to be fully substantively examined.

Despite respectfully requesting for a substantive examination of claim 17 several times previously, the Office again dismissed (see Advisory Action of 04/06/2010) claim 25 as encompassing *"limitations that are similar to claim 17. The claim limitation "packaging the code signed [sic] authorized software component into an installation package" is nothing more than just executable software with signature."*

Claim 25, however, recites:

**packaging the code signed authorized software components into an installation package;**

**configuring install policies to install each code signed authorized executable software component contained in the installation package;**

**configuring certificate rule policies to allow execution of the installed code signed authorized executable software component;**

**configuring enforcement of the policies.**

Not once has the Office applied teachings of the references to the claimed steps of claim 25. This alone, it is respectfully submitted merits returning this application to the Examiner for re-opening the prosecution of this application and reconsideration of the finality of the outstanding Office Action. In this regard, in rejecting claims under 35 U.S.C. Section 103, the Examiner bears the initial burden of presenting a *prima facie* case of obviousness. In re Oetiker, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992).... "A *prima facie* case of obviousness is established when the teachings from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art." In re Bell, 991 F.2d 781, 782, 26 USPQ2d 1529, 1531 (Fed. Cir. 1993) (quoting In re Rinehart, 531 F.2d 1048, 1051, 189 USPQ 143, 147 (CCPA 1976)). Because the claimed steps of the Applicants' claimed invention have

not been shown to be disclosed in the combination of applied references, the undersigned respectfully submits that no *prima facie* case of obviousness has been established. Instead, the claimed steps have been analogized to “*just executable software with signature*” and dismissed out-of-hand, thereby failing to set forth the required *prima facie* case of obviousness as a ground for denying the applicants a patent on the invention defined by claim 25, despite the undersigned’s repeated requests.

In particular, the Office has not pointed where, in the applied combination of references the recited steps of “configuring install policies...; configuring certificate rule policies...; configuring enforcement of the policies” are taught or suggested. These positively recited steps are not found in claim 17 and are not “*similar to*” the steps recited in claim 17 or in any of the other independent claims. As such, they merit a considered substantive examination (indeed, that is what the applicant paid for) and not an out-of-hand dismissal that they are “*similar to*” the steps of other claims when the claimed steps, in fact, have no counterparts in any of the other pending claims.

It is, therefore, respectfully requested that the Board either allow claim 17 or remand this application back to the Examiner for a proper substantive examination of claim 17.

**II. Rejections under 35 U.S.C. §103(a) over Gunvakti et al. in view of Yip et al. in view of Fieres et al. in view of Lambert et al.**

**Independent Claim 17:**

**The Office’s Interpretation of Gunvakti et al. with Respect to Claim 17 is Factually Incorrect**

Claim 17 recites “producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, each software component subject to receiving certification including a unique

identifier; code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate”

Gunyakti et al. does not teach generating a separate and unique PKI certificate for each executable software component, nor does Gunyakti use generated PKI certificates to code sign each of the different executable software components within each gaming machine. Instead, Gunyakti et al. generates a volume license for a number of products and it is this volume license that is signed with a private key to generate the license file 224 – see paragraph [0027]. Therefore, it is the volume license itself that is signed with a private key and NOT “each of the plurality of executable software components”, as required and claimed. In the Advisory Action, the Examiner states “*Therefore, each unique software associated with unique enterprise specific VLK for a plurality of users*”. However, the claims do not recite that each software is associated with a unique volume license – for one or a plurality of users. The claims require “a separate and unique PKI certificate for each ... executable software component” and “code signing each executable software component ... with its respective separate and unique PKI certificate.”. As claimed, each executable software components is code signed with its associated “separate and unique” PKI certificate. In direct contrast, in Gunyakti, it is the license to use the software that is signed, and not the software components themselves, as in the claimed embodiments. This factual error represents yet another independent ground for allowing this application or re-opening the prosecution thereof, as appropriate.

#### **The Office’s Interpretation of Yip et al. is Also Factually Incorrect**

The Office relies upon Yip for the same teaching of producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, and points to Yip’s Figs. 2 and 3 and paragraphs [0048]



and [0046].

In Yip, a conventional Certificate Authority (CA) issues a certificate 106 and an application-specific CA issues a corresponding application-specific certificate 206. See paragraph [0042]. The certificate 106 and application certificate 206 are linked, such that when the certificate 106 is revoked, the application-specific certificates are also preferably revoked. See paragraph [0044]. Thus, the application-specific certificate 206 is a “companion” to the certificate 106.

Note, however, that claim 17 recites:

code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate, each respective PKI certificate being uniquely identified at least by a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate, (underlining for emphasis)

As the application-specific certificate 204 is “for use with the particular application 201”, it necessarily follows that identical executable software components in different ones of the plurality of gaming machines, in Yip, would be associated with different PKI certificates, as each subscriber would receive a different certificate 106 and corresponding different application-specific certificates 206. There is no teaching or suggestion in Yip otherwise.

Indeed, Yip teaches away from the claimed embodiments in which identical application-specific certificates are provided for identical executable software components in different machines. In other words, the CA in Yip would not issue identical certificates 106 to more than one machine/user nor would the CA issue identical companion application-specific certificates 206

to more than on machine/user (subscriber), as each certificate 106 is different and as the application-specific certificates 206 are companions to such different certificates 106. Since the application-specific certificates 206 are companions to a unique subscriber 106, different subscribers using the same software would necessarily be issued different application specific companion certificates 206, which teaches away from the claimed embodiment.

Therefore, since each “particular” application 201 receives a different certificate in Yip, there are believed to be no grounds for holding that Yip teaches or suggests (either alone or in combination with any or all of the other three applied references), the claimed limitation:

**identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates**

**The Combination of Gunvakti and Yip Does Not Teach or Suggest the Claimed Embodiment**

Therefore, the combination of Gunyakti and Yip does not teach or suggest the claim limitations (contrary to that stated in the advisory action of 4/6/10, beginning at line 7), but instead would teach a PKI signed volume license (as taught by Gunyakti) in combination with application-specific certificates in which each particular application received its own certificate (as taught by Yip), with identical executable software components on different gaming machines receiving different application-specific certificates 206 from the application specific CA 204, as again taught by Yip, which combination suggests nothing of the claimed embodiments and teaches away from any embodiment in which “**identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates**”, as claimed herein.

**Fieres Does not Remedy the Shortcomings of Gunvakti-Yip**

The applied reference to Fieres teaches the issuance of application certificates to insure that applications operate at the proper cryptographic level granted for that application by an application domain authority 22. See, for example, Col. 7, lines 44-56 of Fieres. However, there is no teaching or suggestion in Fieres that “identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates”. Nor is there any teaching or suggestion in Fieres that **“non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates”**, as claimed herein.

Fieres does not teach or suggest that **“no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate”**, as claimed herein – nor has the Office identified where such teachings or suggestions may be found. In fact, there is no teaching or suggestion, in the context of the distribution of cryptographic capabilities, that Fieres would allow identical executable components in different machines to have identical certificates, as required herein. Such would surely defeat the security measures. A general allegation that Fieres teaches application certificates with application IDs (see Advisory Action) does not, without more, rise to the level of teaching or suggesting the aforementioned claim limitations, whether considered singly or in combination with the Gunyakti and/or Yip applied references.

#### **Lambert Also Does not Provide the Missing Teachings or Suggestions**

Lastly, **Lambert** was relied on for a teaching of *“a method and system for securely control software execution by identifying and classifying software and locating a rule and associated security level for executing executable software”* (Advisory action of 4/6/2010). However, the

pending claims do not **a)** identify software, **b)** classify software, **c)** locate a rule and/or **d)** locate an associated security level. More to the point, with respect to what **is actually** claimed, Lambert does not teach a software restriction policy (SRP) certificate rule for each executable software component. Quite to the contrary, Lambert teaches **one rule for an entire security level** for executing executable software (see Abstract, lines 3-4). This means that executable software, in Lambert et al. are associated with different security levels, and the rule for that security level may allow or disallow execution thereof. Lambert also teaches a hierarchy of rules, to help distinguish which rule to use, should a piece of software have multiple classifications (see Abstract, last sentence and the “precedence mechanism” to establish which rule applies taught at Col. 3, lines 48-50). In Column 15, lines 29-36, Lambert teaches how rules are selected and at lines 15-20 Lambert describes how rules determine the execution of the file. Lambert, therefore, does not teach or suggest any step of configuring a software restriction policy certificate rule for each of the plurality of executable software components and enforcing each SRP to allow execution of only those software components that is determined to be authorized, as, as claimed by claim 17:

configuring a software restriction policy certificate rule for each of the plurality of executable software components and enforcing each of the software restriction policy certificate rules to allow execution of only those executable software components whose code signed PKI certificate is determined to be authorized.

To the contrary, Lambert et al. teaches away from the claimed embodiments by teaching a one-to-many relationship between the security rules and the executable software components, which is antithetical to the claimed embodiments, which require a software restriction policy for EACH of the plurality of executable software components. The Lambert reference, therefore, does not teach or suggest the subject matter of the claim and does not remedy the shortcomings of Gunyakti, Yip or Fieres.

**The Applied Gunyakti-Yip-Fieres-Lambert Combination Does Not Teach or Suggest the Embodiment Defined by Claim 17**

Indeed, such a four-way combination would teach or suggest, to the person of ordinary skill in the art, PKI signed volume license (as taught by Gunyakti) in combination with application-specific certificates in which each particular application received its own certificate (as taught by Yip), with identical executable software components on different gaming machines receiving different application-specific certificates 206, as again taught by Yip. Software would then be divided amongst different cryptographic levels and application certifications would be issued to insure that applications operate at the proper Cryptographic (Fieres) and Security (Lambert) level granted for that application by an application domain authority. It is respectfully submitted, therefore, that the applied four-way combination, therefore, does not teach or suggest the embodiment of claim 17.

Therefore, it is respectfully requested that the Board reconsider the outstanding rejection of claim 17 and reverse.

**Independent Claim 20**

**The Office's Interpretation of Gunyakti et al. with respect to claim 20 is Factually Incorrect**

Claim 20 recites

“code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;”

**Gunyakti et al.** does not teach code signing each authorized software component with a

PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PCKI certificates and such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate. In fact, Gunyakti does not even generate PKI certificates to code sign each of the different executable software components within each gaming machine. Instead, Gunyakti et al. generates a volume license for a number of products and it is this volume license that is signed with a private key to generate the license file 224 -- see paragraph [0027]. Therefore, it is the volume license itself that is signed with a private key and NOT "each authorized software component", as required and claimed in claim 20. In the Advisory Action, the Examiner states *"Therefore, each unique software associated with unique enterprise specific VLK for a plurality of users"*. However, claim 20 does not recite that each software is associated with a unique volume license -- for one or a plurality of users. Claim 20 requires "code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates ... and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate." Gunyakti's volume license does not teach or suggest any such code signing step. As claimed, **"non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates"**. In direct contrast, in Gunyakti, it is the license to use the software that is signed, and not the software components themselves, as in the claimed embodiments. This factual error represents an independent ground for allowing this application or re-opening the

prosecution thereof, as appropriate.

**The Office's Interpretation of Yip et al. relative to Claim 20 is Also Factually Incorrect**

The Office relies upon **Yip** for the same teaching of producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, and points to Yip's Figs. 2 and 3 and paragraphs [0048] and [0046].

In Yip, a conventional Certificate Authority (CA) issues a certificate 106 to a subscriber and an application-specific CA issues a corresponding application-specific certificate 206. See paragraph [0042]. The certificate 106 and application certificate 206 are linked, such that when the subscriber certificate 106 is revoked, the application-specific certificates 206 are also preferably revoked. See paragraph [0044]. Thus, the application-specific certificate 206 is a "companion" to the subscriber certificate 106.

Note, however, that claim 20 recites:

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate (underlining for emphasis)

As Yip's application-specific certificate 206 is "for use with the particular application 201", it necessarily follows that identical executable software components in different ones of the plurality of gaming machines, in Yip, would be associated with different PKI certificates, as each subscriber would receive a different certificate 106 and corresponding different application-specific

certificates 206. There is no teaching or suggestion in Yip otherwise.

Indeed, Yip teaches away from the claimed embodiments in which **identical application-specific certificates** are provided for identical executable software components in different machines. In other words, the CA in Yip would not issue identical certificates 106 to more than one machine/user nor would the CA issue identical companion application-specific certificates 206 to more than one machine/user, as each certificate 106 is different and as the application-specific certificates 206 are companions to such different certificates 106.

Therefore, since each “particular” application 201 receives a different certificate in Yip (as it must, since the application specific certificate 206 is companion to a unique subscriber certificate 106), there are believed to be no grounds for holding that Yip teaches or suggests (either alone or in combination with any or all of the other applied references), the claimed limitation “code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates”

**The Combination of Gunyakti and Yip Does Not Teach or Suggest the Embodiment of Claim 20**

Therefore, the combination of Gunyakti and Yip does not yield the claim limitation (contrary to that stated in the advisory action of 4/6/10, beginning at line 7), but instead would teach a PKI signed volume license (as taught by Gunyakti) in combination with application-specific certificates in which each particular application received its own certificate (as taught by Yip), with identical executable software components on different gaming machines receiving different application-specific certificates 206, as again taught by Yip, which combination suggests nothing of the claimed embodiments and teaches away from any embodiment in which “**identical**



authorized software components in different ones of the constituent computers are code signed with identical PKI certificates”, as claimed herein.

**Fieres Does not Remedy The Fundamental Shortcomings of Gunvakti-Yip**

The applied reference to **Fieres** teaches the issuance of application certifications to insure that applications operate at the proper cryptographic level granted for that application by an application domain authority 22. However, there is no teaching or suggestion in Fieres that “**identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates**”. Nor is there any teaching or suggestion in Fieres that “**non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates**”, as claimed herein.

Fieres does not teach or suggest that “**non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates**”, as also claim in independent claim 20 – nor has the Office identified where such teachings or suggestions may be found. In fact, there is no teaching or suggestion, in the context of the distribution of cryptographic capabilities, that Fieres would allow identical executable components in different machines to have identical certificates, as required herein. Such would surely defeat the security measures. A general allegation that Fieres teaches application certificates with application IDs (see Advisory Action) does not, without more, rise to the level of teaching the aforementioned claim limitations, whether considered singly or in combination with the Gunvakti and/or Yip applied references.

**Lambert Also Does not Provide the Missing Teachings or Suggestions**

Lastly, **Lambert** was relied on for a teaching of “*a method and system for securely control software execution by identifying and classifying software and locating a rule and associated security level for executing executable software*” (Advisory action of 4/6/2010). However, the pending claims do not **a)** identify software, **b)** classify software, **c)** locate a rule and/or **d)** locate an associated security level. More to the point, with respect to what **is actually** claimed, Lambert does not teach:

configuring a separate software restriction policy for each authorized software component in each of the constituent computers of the gaming system, and associating the configured separate software restriction policy with the PKI certificate with which the authorized software component was code signed;

enforcing the associated software restriction policy for each code signed authorized software component such that each code signed authorized software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy.

Quite to the contrary, Lambert teaches **one rule for an entire security level** for executing executable software (see Abstract, lines 3-4). This means that executable software, in Lambert et al. are associated with different security levels, and the rule for that security level may allow or disallow execution thereof. Lambert also teaches a precedence mechanism and a hierarchy of rules, to help distinguish which rule to use, should a piece of software have multiple classifications (see Abstract, last sentence). In Column 15, lines 29-36, Lambert teaches how rules are selected and at lines 15-20 describes how rules determine the execution of the file. If there was one rule for each software component, such precedence and hierarchical rules would be unnecessary. Lambert, therefore, does not teach or suggest any step of configuring a separate software restriction policy certificate rule for each authorized software component in each of the constituent computers of the gaming system, as claimed herein.

To the contrary, Lambert et al. teaches away from the claimed embodiments by teaching a

one-to-many relationship between the security rules and the executable software components, which is antithetical to the claimed embodiments, which require a software restriction policy to be configured for each authorized software component in each of the constituent computers of the gaming system. The Lambert reference, therefore, does not teach or suggest the subject matter of the claim and does not remedy the shortcomings of Gunyakti, Yip or Fieres.

**The Applied Gunyakti-Yip-Fieres-Lambert Combination Does Not Teach or Suggest the Embodiment Defined by Claim 20**

Indeed, such a four-way combination would teach or suggest, to the person of ordinary skill in the art, PKI signed volume license (as taught by Gunyakti) in combination with application-specific certificates in which each particular application received its own certificate (as taught by Yip), with identical executable software components on different gaming machines receiving different application-specific certificates 206, as again taught by Yip. Software would then be divided amongst different cryptographic levels and application certifications would be issued to insure that applications operate at the proper Cryptographic (Fieres) and Security (Lambert) level granted for that application by an application domain authority. It is respectfully submitted, therefore, that the applied four-way combination, therefore, does not teach or suggest the embodiment of claim 20.

Wholly unsuggested by the applied combination would be the steps of claim 20; namely:

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;

configuring a separate software restriction policy for each authorized software component in each of the constituent computers of the

gaming system, and associating the configured separate software restriction policy with the PKI certificate with which the authorized software component was code signed;

enforcing the associated software restriction policy for each code signed authorized software component such that each code signed authorized software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy.

In view of the foregoing, therefore, it is respectfully requested that the Board reconsider and reverse the obviousness rejection of claim 20.

#### **Independent Claim 24**

#### **The Office's Interpretation of Gunyakti et al. with Respect to Claim 24 is Also Factually Incorrect**

Claim 24 recites

...

producing a separate and unique PKI certificate for each of the plurality of executable software components within the gaming system subject to receive certification, each respective PKI certificate being associated with a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate;

code signing each software component subject to receive certification with its respective separate and unique PKI certificate;

...

**Gunyakti et al.** does not teach the generation of a **separate and unique** PKI certificate for each executable software component subject to receiving certification, as claimed and required herein. As previously noted, **Gunyakti et al.** instead generates a **volume license** for a number of products and it is this volume license that is signed with a private key to generate the license file

224 — see paragraph [0027]. Therefore, it is the volume license itself that is signed with a private key and NOT “each of the plurality of executable software components”, as recited in claim 24. In the Advisory Action, the Examiner states “*Therefore, each unique software associated with unique enterprise specific VLK for a plurality of users*”. However, claim 24 does not recite that each software is associated with a unique volume license — for one or a plurality of users. Instead, claim 24 requires “a separate and unique PKI certificate for each ... executable software component” and “code signing each executable software component ... with its respective separate and unique PKI certificate.” As claimed, each executable software components is code signed with its “separate and unique” PKI certificate. In direct contrast, in Gunyakti, it is the license to use the software that is signed, and not the software components themselves, as in the claimed embodiments. This factual error represents an independent ground for allowing this application or re-opening the prosecution thereof, as appropriate.

**The Office’s Interpretation of Yip et al. relative to claim 24 is Also Factually Incorrect**

The Office relies upon **Yip** for the same teaching of producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, and points to Yip’s Figs. 2 and 3 and paragraphs [0048] and [0046].

As outlined above, in Yip, a conventional Certificate Authority (CA) issues a certificate 106 and an application-specific CA issues a corresponding application-specific certificate 206. See paragraph [0042]. The certificate 106 and application certificate 206 are linked, such that when the certificate 106 is revoked, the application-specific certificates 206 are also preferably revoked. See paragraph [0044]. Thus, the application-specific certificate 206 is a “companion” to the certificate 206.

Note, however, that claim 24 recites:

...  
producing a separate and unique PKI certificate for each of the plurality of executable software components within the gaming system subject to receive certification, each respective PKI certificate being associated with a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate; (underlining for emphasis)

As the application-specific certificate 206 is “for use with the particular application 201”, it necessarily follows that identical executable software components in different ones of the plurality of gaming machines, in Yip, would be associated with different PKI certificates, as each subscriber would receive a different certificate 106 and corresponding different application-specific certificates 206. There is no teaching or suggestion in Yip otherwise.

Indeed, Yip teaches away from the claimed embodiments in which identical application-specific certificates are provided for identical executable software components in different machines. In other words, the CA in Yip would not issue identical certificates 106 to more than one machine/user nor would the CA issue identical companion application-specific certificates 206 to more than one machine/user, as each certificate 106 is different and as the application-specific certificates 206 are companions to such different certificates 106.

Therefore, since each “particular” application 201 receives a different certificate in Yip, there are believed to be no grounds for holding that Yip teaches or suggests (either alone or in combination with any or all of the other three applied references), the claimed limitation:

such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates

**The Combination of Gunyakti and Yip Does Not Teach or Suggest the Claimed Embodiment**

Therefore, the combination of Gunyakti and Yip does not yield the claim limitations (contrary to that stated in the advisory action of 4/6/10, beginning at line 7), but instead would teach a PKI signed volume license (as taught by Gunyakti) in combination with application-specific certificates in which each particular application received its own certificate (as taught by Yip), with identical executable software components on different gaming machines receiving different application-specific certificates 206, as again taught by Yip, which combination suggests nothing of the claimed embodiments and teaches away from any embodiment in which “**identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates**”, as recited in independent claim 24.

**Fieres Does not Remedy The Fundamental Shortcomings of Gunyakti-Yip**

The applied reference to **Fieres** teaches the issuance of application certifications to insure that applications operate at the proper cryptographic level granted for that application by an application domain authority 22. However, there is no teaching or suggestion in Fieres that “identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates”. Nor is there any teaching or suggestion in Fieres that “**non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates**”, as claimed herein.

Fieres also does not teach or suggest that “**no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate**”, as claimed herein – nor has the Office identified where such teachings or suggestions may be found. In fact, there is no teaching or suggestion, in the context of the distribution of cryptographic capabilities, that Fieres would allow identical executable components in different machines to have identical certificates, as required herein. Such would surely defeat the security measures. A general allegation that Fieres teaches application certificates with application IDs (see Advisory Action) does not, without more, rise to the level of teaching the aforementioned claim limitations, whether considered singly or in combination with the Gunyakti and/or Yip applied references.

**Lambert Also Does not Provide the Missing Teachings or Suggestions**

Lastly, **Lambert** was relied on for a teaching of “*a method and system for securely control software execution by identifying and classifying software and locating a rule and associated security level for executing executable software*” (Advisory action of 4/6/2010). However, the pending claims do not **a)** identify software, **b)** classify software, **c)** locate a rule and/or **d)** locate an associated security level. More to the point, with respect to what **is actually** claimed, Lambert does not teach any steps of configuring or enforcing a certificate software restriction policy (SRP) for each of the respective separate and unique PKI certificates, as recited in claim 24. Quite to the contrary, Lambert teaches **one rule for an entire security level** for executing executable software (see Abstract, lines 3-4). This means that executable software, in Lambert et al. are associated with different security levels, and the rule for that security level may allow or disallow execution thereof. Lambert also teaches a precedence mechanism and hierarchy of rules, to help distinguish which rule to use, should a piece of software have multiple classifications (see Abstract, last sentence). In Column 15, lines 29-36, Lambert teaches how rules are selected and at lines 15-20



describes how rules determine the execution of the file, which would be unnecessary if there was one rule for each software component, as claimed herein. In Lambert, therefore, does not teach or suggest any step of configuring and enforcing a certificate software restriction policy for each of the respective separate and unique PKI certificates, as, as recited in claim 24.

To the contrary, Lambert et al. teaches away from the claimed embodiments by teaching a one-to-many relationship between the security rules and the executable software components, which is antithetical to the embodiment of claim 24, which requires the configuration and enforcement of a certificate software restriction policy for EACH of the plurality of separate and unique PKI certificates. The Lambert reference, therefore, does not teach or suggest the subject matter of the claim and does not remedy the shortcomings of Gunyakti, Yip or Fieres.

**The Applied Gunyakti-Yip-Fieres-Lambert Combination Does Not Teach or Suggest the Embodiment Defined by Claim 24**

Indeed, such a four-way combination would teach or suggest, to the person of ordinary skill in the art, PKI signed volume license (as taught by Gunyakti) in combination with application-specific certificates in which each particular application received its own certificate (as taught by Yip), with identical executable software components on different gaming machines receiving different application-specific certificates 204, as again taught by Yip. Software would then be divided amongst different cryptographic levels and application certifications would be issued to insure that applications operate at the proper Cryptographic (Fieres) and Security (Lambert) level granted for that application by an application domain authority. It is respectfully submitted, therefore, that the applied four-way combination, therefore, does not teach or suggest the embodiment of claim 24.

Moreover, claim 24 also recites:

code signing each software component subject to receive certification with its respective separate and unique PKI certificate;

configuring a certificate software restriction policy for each of the respective separate and unique PKI certificates, and

enforcing the certificate software restriction policy for each of the respective separate and unique PKI certificates.

In contradistinction, the primary reference to Gunyakti advocates volume licenses (how can a volume license be interpreted as a “separate and unique PKI certificate for each of the plurality of executable software components”?), Yip advocates companion application-specific certificates and Lambert calls for a hierarchy of rules to enable the application of a specific rule to a specific application. The applied combination does not teach code signing each software component subject to receive certification with its respective separate and unique PKI certificate (compare to Gunyakti’s volume licenses), configuring a certificate software restriction policy for each of the respective separate and unique PKI certificates (compare with the one-to-many relationship of Lambert’s rules to the applications) or enforcing the certificate software restriction policy for each of the respective separate and unique PKI certificates, as claimed herein.

None of the applied references, alone or in combination, teach or suggest the claimed embodiments. The prior art (Yip) teaches that each “particular” software is signed with a different application specific certificate 206 that is companion to a unique subscriber certificate 106. The prior art (Gunyakti) also teaches code signing volume licenses ( $\neq$  executable software components). The prior art also teaches security levels (Lambert) or cryptographic levels and rules (Fieres) that may or may not allow execution of software components. It is respectfully submitted that the claimed elements are most assuredly not combined “*according to known methods*”, as the Office asserts – nor would any combination of these references teach, suggest or result in the claimed embodiments. Therefore, reversal of the obviousness rejection applied to claim 24 is also

respectfully requested.

**I. Rejections under 35 U.S.C. §103(a) over Lambert et al. in view of Gunyakti et al. in view of Yip et al.**

**Independent Claim 22**

Lambert teaches rules associated with security levels and teaches rules based upon the classification of its folder or based upon its fully qualified path. See for example, the Abstract, Col. 3, lines 30-34 and Column 13 of Lambert.

However, it is respectfully submitted that a teaching of a rule based on the classification of a file based on its specific folder or based upon its fully qualified path, even when considered in combination with the above-discussed teachings and suggestions of Gunyakti and Yip, does not rise to the level of a teaching or suggestion of claim 22 as a whole.

Indeed, even if the teachings and suggestions of Lambert, Gunyakti and Yip were known to a person of ordinary skill in the art, such a person would not be successful in developing the embodiment claimed in method of claim 22.

Indeed, missing from the applied combination are claim 22's steps of:

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;

Indeed, **Gunyakti et al.** does not teach code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed

with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate. Instead, Gunyakti et al. generates a volume license for a number of products and it is this volume license that is signed with a private key to generate the license file 224 – see paragraph [0027]. Therefore, it is the volume license itself that is signed with a private key and NOT “each of the plurality of executable software components”, as required and claimed. In the Advisory Action, the Examiner states “*Therefore, each unique software associated with unique enterprise specific VLK for a plurality of users*”. However, claim 22 does not recite that each software is associated with a unique volume license – for one or a plurality of users – even if Gunyakti taught such, which it does not. Claim 22 claims that each authorized software component is code signed with a PKI certificate such that the claimed conditions (“such that ... such that ... and such that ...”) hold true across constituent computers of the gaming system. It is respectfully submitted that Gunyakti’s volume license does not meet the claim requirements.

The Office relies upon **Yip** for the same teaching of producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, and points to Yip’s Figs. 2 and 3 and paragraphs [0048] and [0046].

As noted above, in Yip, a conventional Certificate Authority (CA) issues a certificate 106 and an application-specific CA issues a corresponding application-specific certificate 206. See paragraph [0042]. The subscriber certificate 106 and application-specific certificate 206 are tightly bound, such that when the certificate 106 is revoked, the application-specific certificates 206 are also preferably revoked. See paragraph [0044]. Thus, the application-specific certificate 206 is a “companion” to the certificate 106.

Note, however, that claim 22 recites:

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate; (underlining for emphasis)

As the application-specific certificate 206 is “for use with the particular application 201”, it necessarily follows that identical executable software components in different ones of the plurality of gaming machines, in Yip, would be associated with different PKI certificates, as each subscriber would receive a different certificate 106 and corresponding different application-specific certificates 206. There is no teaching or suggestion in Yip otherwise.

Indeed, Yip teaches away from the claimed embodiments in which identical application-specific certificates are provided for identical executable software components in different machines. In other words, the CA in Yip would not issue identical certificates 106 to more than one machine/user nor would the CA issue identical companion application-specific certificates 206 to more than one machine/user, as each certificate 106 is different and as the application-specific certificates 206 are companions to such different certificates 106.

Therefore, since each “particular” application 201 receives a different certificate in Yip, there are believed to be no grounds for holding that Yip teaches or suggests (either alone or in combination with Lambert or Gunyakti) the claimed limitation:

identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates

**The Combination of Lambert, Gunyakti and Yip Does Not Teach or Suggest the Claimed Embodiment**

Whereas Lambert teaches security levels and rules for security levels and folder classifications and fully qualified paths, the Gunyakti- Yip combination fails to teach the claimed certificate software restriction policy SRP configuring step. As these references fail to teach the configuration of a separate and unique certificate SRP for each authorized executable software component each of the constituent computers of the gaming system and fails to teach the code signing step as claimed herein (i.e., identical authorized software components in different ones of the constituent computers being code signed with identical PKI certificates, non-identical authorized software components in different ones of the constituent computers being code signed with separate and different PKI certificates and no two non-identical authorized software components in different ones of the constituent gaming machines being code signed with a same PKI certificate), the claimed embodiment as a whole would not have been obvious to a person of ordinary skill in the art.

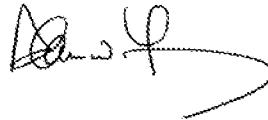
Indeed, the applied combination, at most would teach or suggest a system having a PKI signed volume license (as taught by Gunyakti) and application-specific certificates in which each particular application received its own certificate (as taught by Yip), but with identical executable software components on different gaming machines receiving different application-specific certificates 206, as again taught by Yip. The Lambert teachings of the path rules do not, however, remedy the fundamental shortcomings of Gunyakti and Yip and the applied combination fails to teach or suggest each of the steps of claim 22. It is, therefore, respectfully requested that the Board reconsider the obviousness rejection of claim 22 and reverse.

The dependent claims stand or fall with their respective independent claims.

Applicants' attorney, therefore, respectfully submits to the Board that the outstanding Final Rejection of the claims is in error, that all claims are allowable and that the present application is in condition for immediate allowance and passage to issue.

An oral hearing is not requested.

Respectfully submitted,



Date: August 19, 2010

By:

Alan W. Young  
Attorney for Applicants  
Registration No. 37,970

Young Law Firm, P.C.  
4370 Alpine Rd., Ste. 106  
Portola Valley, CA 94028  
Tel.: (650) 851-7210  
Fax: (650) 851-7232

## IX. CLAIMS APPENDIX

1. **(Withdrawn)** A PKI certificate architecture for a network connected gaming system, the gaming system including a plurality of gaming machines each having a plurality of executable software components, wherein each different executable software component within each gaming machine within the gaming system subject to receive certification is uniquely associated with a unique identifier and is signed with a separate and unique PKI certificate, the separate and unique PKI certificate being uniquely identified at least by the unique identifier, wherein identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are signed with separate and different PKI certificates, and such that no two non-identical executable software components in different gaming machines are signed with a same PKI certificate.

2. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein each software component is authorized by a regulatory authority.

3. **(Withdrawn)** A PKI certificate architecture according to ~~claim 1~~ claim 2, wherein the separate and unique PKI certificate is produced by the certification lab, by the gaming system supplier or by the trusted party designated by the regulatory authority.

4. **(Withdrawn)** A PKI certificate architecture according to ~~claim 1~~ claim 2, wherein each software component is code signed by a certification lab, by a gaming system supplier or by a trusted party designated by the regulatory authority.



5. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein the separate and unique identifier is a certificate field selected from a “Subject” field, an “issued to” field, a “subject name” field, a “CommonName” field, a “provider” field or a “publisher” field.

6. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein the unique identifier comprises at least one of fields and field extensions.

7. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein the unique identifier comprises at least one of a plurality of fields selected from among:

a software component part number;

a software component major version number;

a software component minor version number;

a software component build number;

a software component revision number;

a software component project name;

a software component type of software component;

a software component language variant;

a software component game regulation variant;

a software component friendly name;

an identification of the certification laboratory, and

an identification of the client.

8. **(Withdrawn)** A PKI certificate architecture according to claim 7, wherein the unique identifier is a concatenation of selected identifiers.

9. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is reported in the Windows event log upon execution of the software component.

10. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is reported in the source field of the Windows event log upon execution of the software component.

11. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is reported in the Windows event log upon execution of the software component in a predetermined event log bin upon execution of the software component.

12. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is traceable in at least one of:

source code;

Windows File Properties;

Trusted Inventory;

Windows Event Log;

Software Restriction Policies, and

Certificate Store.

13. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein the network connected gaming system is connected in at least one of a local area system and wide area network.

14. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein the network connected gaming system comprises at least one of gaming terminals, gaming servers and computers.

15. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein the unique identifier contains identification information delimited with file-name-allowed non-alphanumeric characters to facilitate human identification, string searches and file searches.

16. **(Withdrawn)** A PKI certificate architecture according to claim 1, wherein a selected set of identification information making up the unique identifier are used for making up the file name of PKI certificate related files such as \*.CER, \*.P7B and \*.PVK such as to facilitate human identification, string searches and file searches.

17. **(Previously Presented)** A method for a network connected gaming system to prevent unauthorized software components of constituent computers of the gaming system from executing, the gaming system including a plurality of gaming machines each having a plurality of executable software components, the method comprising the steps of:

producing a separate and unique PKI certificate for each of the plurality of executable software components subject to receiving certification within each gaming machine, each software component subject to receiving certification including a unique identifier;

code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate, each respective PKI certificate being uniquely

identified at least by a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate, and

configuring a software restriction policy certificate rule for each of the plurality of executable software components and enforcing each of the software restriction policy certificate rules to allow execution of only those executable software components whose code signed PKI certificate is determined to be authorized.

18. **(Previously Presented)** A method according to claim 17, further comprising the step of configuring software restriction policy rules to prevent execution of unauthorized software components.

19. **(Previously Presented)** A method according to claim 17, further comprising the step of configuring software restriction policy rules to prevent execution of all not explicitly authorized software components.

20. **(Previously Presented)** A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, comprising the steps of:

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;

configuring a separate software restriction policy for each authorized software component in each of the constituent computers of the gaming system, and associating the configured separate software restriction policy with the PKI certificate with which the authorized software component was code signed;

enforcing the associated software restriction policy for each code signed authorized software component such that each code signed authorized software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy.

21. **(Previously Presented)** A method according to claim 20, wherein the authorized software components are mandated by a regulatory body.

22. **(Previously Presented)** A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, comprising the steps of:

configuring a separate and unique certificate software restriction policy for each authorized executable software component of each of the constituent computers of the gaming system such that the each authorized executable software component in each of the constituent

computers of the gaming system must be authorized to run by its associated separate software restriction policy;

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;

configuring a path software restriction policy to prevent unauthorized software components from executing;

configuring a path software restriction policy to prevent non-explicitly authorized software components from executing;

enforcing the certificate software restriction policy configured for each of the code signed authorized executable software components of each of the constituent computers of the gaming system, and

enforcing the path software restriction policies.

23. **(Previously Presented)** A method according to claim 22, wherein the authorized software components are mandated by a regulatory body.

24. **(Previously Presented)** A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, the gaming system including a plurality of gaming machines each having a plurality of executable software components, the method comprising the steps of:

producing a separate and unique PKI certificate for each of the plurality of executable software components within the gaming system subject to receive certification, each respective PKI certificate being associated with a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate;

code signing each software component subject to receive certification with its respective separate and unique PKI certificate;

configuring a certificate software restriction policy for each of the respective separate and unique PKI certificates, and

enforcing the certificate software restriction policy for each of the respective separate and unique PKI certificates.

25. **(Previously Presented)** A method for downloading authorized executable software components and allowing execution of downloaded authorized executable software components of a plurality of gaming machines of a network connected gaming system, comprising the steps of:

for each of the plurality of gaming machines of the network connected gaming system:

code signing each authorized executable software component with a separate PKI certificate that is unique to the authorized software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different gaming machines are code signed with a same PKI certificate;

packaging the code signed authorized software components into an installation package;

configuring install policies to install each code signed authorized executable software component contained in the installation package;

configuring certificate rule policies to allow execution of the installed code signed authorized executable software component;

configuring enforcement of the policies.

26-81. **(Canceled)**

82. **(Withdrawn)** An automated platform to enable an on-going regulatory certification of a plurality of authorized software components of a network connected gaming system including a plurality of computers, the method comprising:

a reference platform representative of a target network connected gaming system and comprising a software-building environment located at a manufacturer or subcontractor of the software components;

a certification platform located at a regulatory certification authority, the certification platform being substantially identical to the reference platform;



code-signing means for enabling the manufacturer or subcontractor to associate a separate and unique PKI certificate with each authorized software component subject to regulatory certification such that identical authorized software components subject to regulatory certification in different ones of the plurality of gaming machines of the network connected gaming system are code signed with identical PKI certificates, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates, and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate, and

a secure communication link between the reference platform and the certification platform for enabling manufacturer or designated subcontractors to remotely configure the software building environment on the certification platform.

83. **(Canceled)**

84. **(Withdrawn)** An automated platform according to claim 82, wherein the authorized software components to be downloaded to the network connected gaming system are tested by the certification laboratory.

85. **(Withdrawn)** An automated platform according to claim 82, wherein the authorized software components to be downloaded to the network connected gaming system are compiled by the certification laboratory.

86. **(Withdrawn)** An automated platform according to claim 82, further comprising a secure communication link between the reference platform and the certification\_for enabling remote assistance.

87. **(Withdrawn)** An automated platform according to claim 82, further comprising a secure communication link between the reference platform and the certification\_for enabling users to carry out certification steps from a remotely located computer.

88. **(Withdrawn)** An automated platform according to claim 82, wherein the code signing means comprises a certificate authority under control of the manufacturer for generating certificates.

89. **(Withdrawn)** An automated platform according to claim 82, wherein the code signing means comprises a certificate authority under control of the regulatory certification authority for generating certificates.

90. **(Withdrawn)** An automated platform according to claim 82, further comprising means for maintaining the software-building environment of the reference platform and the software-building environment of the certification platform synchronized.

91-97. **(Canceled)**

**X. EVIDENCE APPENDIX**

None.

**XI. RELATED PROCEEDINGS APPENDIX**

None.